Borel ranks and Wadge degrees of context free ω-languages

Olivier Finkel

Equipe de Logique Mathématique
U.F.R. de Mathématiques, Université Paris 7
2 Place Jussieu 75251 Paris cedex 05, France
finkel@logique.jussieu.fr.

Abstract. We show that the Borel hierarchy of the class of context free ω-languages, or even of the class of ω-languages accepted by Büchi 1-counter automata, is the same as the Borel hierarchy of the class of ω-languages accepted by Turing machines with a Büchi acceptance condition. In particular, for each recursive non null ordinal α, there exist some $\mathbf{\Sigma}^0_\alpha$-complete and some $\mathbf{\Pi}^0_\alpha$-complete ω-languages accepted by Büchi 1-counter automata. And the supremum of the set of Borel ranks of context free ω-languages is an ordinal γ^1_2 which is strictly greater than the first non recursive ordinal ω^{CK}_1. We then extend this result, proving that the Wadge hierarchy of context free ω-languages, or even of ω-languages accepted by Büchi 1-counter automata, is the same as the Wadge hierarchy of ω-languages accepted by Turing machines with a Büchi or a Muller acceptance condition.

1 Introduction

Languages of infinite words accepted by finite automata were first studied by Büchi to prove the decidability of the monadic second order theory of one successor over the integers. The theory of the so called regular ω-languages is now well established and has found many applications for specification and verification of non-terminating systems; see [Tho90,Sta97,PP04] for many results and references. More powerful machines, like pushdown automata, Turing machines, have also been considered for the reading of infinite words, see Staiger's survey [Sta97] and the fundamental study [EH93] of Engelfriet and Hoogeboom on **X**-automata, i.e. finite automata equipped with a storage type **X**. A way to study the complexity of ω-languages is to study their topological complexity, and particularly to locate them with regard to the Borel and the projective hierarchies. On one side all ω-languages accepted by *deterministic* **X**-automata with a Muller acceptance condition are Boolean combinations of $\mathbf{\Pi}^0_2$-sets hence $\mathbf{\Delta}^0_3$-sets, [Sta97,EH93]. This implies, from Mc Naughton's Theorem, that all regular ω-languages, which are accepted by deterministic Muller automata, are also

$\mathbf{\Delta}_3^0$-sets. On the other side, for *non deterministic* finite machines, the question, posed by Lescow and Thomas in [LT94], naturally arises: what is the topological complexity of ω-languages accepted by automata equipped with a given storage type **X**? It is well known that every ω-language accepted by a Turing machine (hence also by a **X**-automaton) with a Muller acceptance condition is an analytic set. In previous papers, we proved that there are context free ω-languages, accepted by Büchi or Muller pushdown automata, of every finite Borel rank, of infinite Borel rank, or even being analytic but non Borel sets, [DFR01,Fin01c,Fin03a,Fin03b]. In this paper we show that the Borel hierarchy of ω-languages accepted by **X**-automata, for every storage type **X** such that 1-counter automata can be simulated by **X**-automata, is the same as the Borel hierarchy of ω-languages accepted by Turing machines with a Büchi acceptance condition. In particular, for each recursive non null ordinal α, there exist some $\mathbf{\Sigma}_\alpha^0$-complete and some $\mathbf{\Pi}_\alpha^0$-complete ω-languages accepted by Büchi 1-counter automata, hence also in the class $\mathbf{CFL}_\omega$ of context free ω-languages.

We have to indicate here a mistake in the conference paper [Fin05]. We wrote in that paper that it is well known that if $L \subseteq \Sigma^\omega$ is a (lightface) Σ_1^1 set, i.e. accepted by a Turing machine with a Büchi acceptance condition, and is a Borel set of rank α, then α is smaller than the Church Kleene ordinal ω_1^{CK}, which is the first non recursive ordinal. This fact, which is true if we replace the (lightface) class Σ_1^1 by the (lightface) class Δ_1^1, is actually not true. Kechris, Marker and Sami proved in [KMS89] that the supremum of the set of Borel ranks of (lightface) Π_1^1, so also of (lightface) Σ_1^1, sets is the ordinal γ_2^1. This ordinal is precisely defined in [KMS89] and it is strictly greater than the ordinal ω_1^{CK}. The proofs we give in this paper show that the ordinal γ_2^1 is also the supremum of the set of Borel ranks of ω-languages accepted by Büchi 1-counter automata, or of context free ω-languages.

By considering the Wadge hierarchy which is a great refinement of the Borel hierarchy, [Wad83,Dup01], we show the following strengthening of the preceding result. The Wadge hierarchy of the class $\mathbf{r\text{-}BCL}(1)_\omega$ of ω-languages accepted by real time 1-counter Büchi automata, hence also of the class $\mathbf{CFL}_\omega$, is the Wadge hierarchy of the class of ω-languages accepted by Turing machines with a Büchi acceptance condition.

We think that the surprising result obtained in this paper is of interest for both logicians working on hierarchies arising in recursion theory or in descriptive set theory, and also for computer scientists working on questions connected with non-terminating systems, like the construction of effective strategies in infinite games, [Wal00,Tho02,Cac02,Ser04b].

The paper is organized as follows. In Section 2 we define multicounter automata which will be a useful tool in the sequel. Recall on Borel hierarchy is given in Section 3. In Section 4 is studied the Borel hierarchy of ω-languages accepted by real time 8-counter automata. The Borel hierarchy of the class $\mathbf{r\text{-}BCL}(1)_\omega$ is studied in Section 5. Results about the Wadge hierarchy of the class $\mathbf{r\text{-}BCL}(1)_\omega$ are given in Section 6.

2 Multicounter automata

We assume the reader to be familiar with the theory of formal (ω)-languages [Tho90,Sta97]. We shall use usual notations of formal language theory.

When Σ is a finite alphabet, a *non-empty finite word* over Σ is any sequence $x = a_1 \ldots a_k$, where $a_i \in \Sigma$ for $i = 1, \ldots, k$, and k is an integer ≥ 1. The *length* of x is k, denoted by $|x|$. The *empty word* has no letter and is denoted by λ; its length is 0. For $x = a_1 \ldots a_k$, we write $x(i) = a_i$ and $x[i] = x(1) \ldots x(i)$ for $i \leq k$ and $x[0] = \lambda$. $\Sigma^\star$ is the *set of finite words* (including the empty word) over Σ. The *first infinite ordinal* is ω. An ω-*word* over Σ is an ω -sequence $a_1 \ldots a_n \ldots$, where for all integers $i \geq 1$, $\quad a_i \in \Sigma$. When σ is an ω-word over Σ, we write $\sigma = \sigma(1)\sigma(2)\ldots\sigma(n)\ldots$, where for all i, $\sigma(i) \in \Sigma$, and $\sigma[n] = \sigma(1)\sigma(2)\ldots\sigma(n)$ for all $n \geq 1$ and $\sigma[0] = \lambda$.

The *prefix relation* is denoted $\sqsubseteq$: a finite word u is a *prefix* of a finite word v (respectively, an infinite word v), denoted $u \sqsubseteq v$, if and only if there exists a finite word w (respectively, an infinite word w), such that $v = u.w$. The *set of ω-words* over the alphabet Σ is denoted by Σ^ω. An ω-*language* over an alphabet Σ is a subset of Σ^ω. The complement (in Σ^ω) of an ω-language $V \subseteq \Sigma^\omega$ is $\Sigma^\omega - V$, denoted V^-.

Definition 1. *Let k be an integer ≥ 1. A k-counter machine (k-CM) is a 4-tuple $\mathcal{M} = (K, \Sigma, \Delta, q_0)$, where K is a finite set of states, Σ is a finite input alphabet, $q_0 \in K$ is the initial state, and $\Delta \subseteq K \times (\Sigma \cup \{\lambda\}) \times \{0,1\}^k \times K \times \{0,1,-1\}^k$ is the transition relation. The k-counter machine $\mathcal{M}$ is said to be real time iff: $\Delta \subseteq K \times \Sigma \times \{0,1\}^k \times K \times \{0,1,-1\}^k$, i.e. iff there are not any λ-transitions. If the machine $\mathcal{M}$ is in state q and $c_i \in \mathbf{N}$ is the content of the ith counter $\mathcal{C}_i$ then the configuration (or global state) of $\mathcal{M}$ is the $(k+1)$-tuple $(q, c_1, \ldots, c_k)$.*

For $a \in \Sigma \cup \{\lambda\}$, $q, q' \in K$ and $(c_1, \ldots, c_k) \in \mathbf{N}^k$ such that $c_j = 0$ for $j \in E \subseteq \{1, \ldots, k\}$ and $c_j > 0$ for $j \notin E$, if $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_k) \in \Delta$ where $i_j = 0$ for $j \in E$ and $i_j = 1$ for $j \notin E$, then we write:

$$a : (q, c_1, \ldots, c_k) \mapsto_{\mathcal{M}} (q', c_1 + j_1, \ldots, c_k + j_k)$$

$\mapsto_{\mathcal{M}}^\star$ is the transitive and reflexive closure of $\mapsto_{\mathcal{M}}$. (The subscript $\mathcal{M}$ will be omitted whenever the meaning remains clear).

Thus we see that the transition relation must satisfy:
if $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_k) \in \Delta$ and $i_m = 0$ for some $m \in \{1, \ldots, k\}$, then $j_m = 0$ or $j_m = 1$ (but j_m may not be equal to -1).

Let $\sigma = a_1 a_2 \ldots a_n$ be a finite word over Σ. An sequence of configurations $r = (q_i, c_1^i, \ldots c_k^i)_{1 \leq i \leq p}$, for $p \geq n + 1$, is called a run of $\mathcal{M}$ on σ, starting in configuration $(p, c_1, \ldots, c_k)$, iff:

(1) $(q_1, c_1^1, \ldots c_k^1) = (p, c_1, \ldots, c_k)$
(2) for each $i \geq 1$, there exists $b_i \in \Sigma \cup \{\lambda\}$ such that $b_i : (q_i, c_1^i, \ldots c_k^i) \mapsto_{\mathcal{M}} (q_{i+1}, c_1^{i+1}, \ldots c_k^{i+1})$

(3) $a_1.a_2.a_3 \ldots a_n = b_1.b_2.b_3 \ldots b_p$

Let $\sigma = a_1 a_2 \ldots a_n \ldots$ be an ω-word over Σ. An ω-sequence of configurations $r = (q_i, c_1^i, \ldots c_k^i)_{i \geq 1}$ is called a run of $\mathcal{M}$ on σ, starting in configuration $(p, c_1, \ldots, c_k)$, iff:

(1) $(q_1, c_1^1, \ldots c_k^1) = (p, c_1, \ldots, c_k)$
(2) for each $i \geq 1$, there exists $b_i \in \Sigma \cup \{\lambda\}$ such that $b_i : (q_i, c_1^i, \ldots c_k^i) \mapsto_{\mathcal{M}} (q_{i+1}, c_1^{i+1}, \ldots c_k^{i+1})$ such that either $\quad a_1 a_2 \ldots a_n \ldots = b_1 b_2 \ldots b_n \ldots$
or $\quad b_1 b_2 \ldots b_n \ldots$ is a finite prefix of $\quad a_1 a_2 \ldots a_n \ldots$

The run r is said to be complete when $a_1 a_2 \ldots a_n \ldots = b_1 b_2 \ldots b_n \ldots$
For every such run, $\mathrm{In}(r)$ is the set of all states entered infinitely often during run r.
A complete run r of M on σ, starting in configuration $(q_0, 0, \ldots, 0)$, will be simply called "a run of M on σ".

Definition 2. *A Büchi k-counter automaton is a 5-tuple $\mathcal{M} = (K, \Sigma, \Delta, q_0, F)$, where $\mathcal{M}' = (K, \Sigma, \Delta, q_0)$ is a k-counter machine and $F \subseteq K$ is the set of accepting states. The ω-language accepted by $\mathcal{M}$ is*

$$L(\mathcal{M}) = \{\sigma \in \Sigma^\omega \mid \text{ there exists a run } r \text{ of } \mathcal{M} \text{ on } \sigma \text{ such that } \mathrm{In}(r) \cap F \neq \emptyset\}$$

Definition 3. *A Muller k-counter automaton is a 5-tuple $\mathcal{M} = (K, \Sigma, \Delta, q_0, \mathcal{F})$, where $\mathcal{M}' = (K, \Sigma, \Delta, q_0)$ is a k-counter machine and $\mathcal{F} \subseteq 2^K$ is the set of accepting sets of states. The ω-language accepted by $\mathcal{M}$ is*

$$L(\mathcal{M}) = \{\sigma \in \Sigma^\omega \mid \text{ there exists a run } r \text{ of } \mathcal{M} \text{ on } \sigma \text{ such that } \exists F \in \mathcal{F} \ \mathrm{In}(r) = F\}$$

The class of Büchi k-counter automata will be denoted **BC**(k).
The class of real time Büchi k-counter automata will be denoted **r-BC**(k).
The class of ω-languages accepted by Büchi k-counter automata will be denoted **BCL**$(k)_\omega$.
The class of ω-languages accepted by real time Büchi k-counter automata will be denoted **r-BCL**$(k)_\omega$.

It is well known that an ω-language is accepted by a (real time) Büchi k-counter automaton iff it is accepted by a (real time) Muller k-counter automaton [EH93]. Notice that it cannot be shown without using the non determinism of automata and this result is no longer true in the deterministic case.
Remark that 1-counter automata introduced above are equivalent to pushdown automata whose stack alphabet is in the form $\{Z_0, A\}$ where Z_0 is the bottom symbol which always remains at the bottom of the stack and appears only there and A is another stack symbol. The pushdown stack may be seen like a counter whose content is the integer N if the stack content is the word $Z_0.A^N$.
In the model introduced here the counter value cannot be increased by more than 1 during a single transition. However this does not change the class of ω-languages accepted by such automata. So the class **BCL**$(1)_\omega$ is equal to the class **1-ICL**$_\omega$, introduced in [Fin01d], and it is a strict subclass of the class **CFL**$_\omega$ of context free ω-languages accepted by Büchi pushdown automata.

3 Borel hierarchy

We assume the reader to be familiar with basic notions of topology which may be found in [Mos80,LT94,Kec95,Sta97,PP04]. There is a natural metric on the set Σ^ω of infinite words over a finite alphabet Σ which is called the *prefix metric* and defined as follows. For $u, v \in \Sigma^\omega$ and $u \neq v$ let $\delta(u, v) = 2^{-l_{\mathrm{pref}(u,v)}}$ where $l_{\mathrm{pref}(u,v)}$ is the first integer n such that the $(n + 1)^{st}$ letter of u is different from the $(n+1)^{st}$ letter of v. This metric induces on Σ^ω the usual Cantor topology for which *open subsets* of Σ^ω are in the form $W.\Sigma^\omega$, where $W \subseteq \Sigma^\star$. A set $L \subseteq \Sigma^\omega$ is a *closed set* iff its complement $\Sigma^\omega - L$ is an open set. Define now the *Borel Hierarchy* of subsets of Σ^ω:

Definition 4. *For a non-null countable ordinal α, the classes $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$ of the Borel Hierarchy on the topological space Σ^ω are defined as follows:*
$\mathbf{\Sigma}^0_1$ is the class of open subsets of Σ^ω, $\mathbf{\Pi}^0_1$ is the class of closed subsets of Σ^ω, and for any countable ordinal $\alpha \geq 2$:
$\mathbf{\Sigma}^0_\alpha$ is the class of countable unions of subsets of Σ^ω in $\bigcup_{\gamma < \alpha} \mathbf{\Pi}^0_\gamma$.
$\mathbf{\Pi}^0_\alpha$ is the class of countable intersections of subsets of Σ^ω in $\bigcup_{\gamma < \alpha} \mathbf{\Sigma}^0_\gamma$.

For a countable ordinal α, a subset of Σ^ω is a Borel set of *rank α* iff it is in $\mathbf{\Sigma}^0_\alpha \cup \mathbf{\Pi}^0_\alpha$ but not in $\bigcup_{\gamma < \alpha} (\mathbf{\Sigma}^0_\gamma \cup \mathbf{\Pi}^0_\gamma)$.

There are also some subsets of Σ^ω which are not Borel. In particular the class of Borel subsets of Σ^ω is strictly included into the class $\mathbf{\Sigma}^1_1$ of *analytic sets* which are obtained by projection of Borel sets, see for example [Sta97,LT94,PP04,Kec95] for more details.
We now define completeness with regard to reduction by continuous functions. For a countable ordinal $\alpha \geq 1$, a set $F \subseteq \Sigma^\omega$ is said to be a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$, $\mathbf{\Sigma}^1_1$)-*complete set* iff for any set $E \subseteq Y^\omega$ (with Y a finite alphabet): $E \in \mathbf{\Sigma}^0_\alpha$ (respectively, $E \in \mathbf{\Pi}^0_\alpha$, $E \in \mathbf{\Sigma}^1_1$) iff there exists a continuous function $f : Y^\omega \to \Sigma^\omega$ such that $E = f^{-1}(F)$. $\mathbf{\Sigma}^0_n$ (respectively $\mathbf{\Pi}^0_n$)-complete sets, with n an integer ≥ 1, are thoroughly characterized in [Sta86].
The (lightface) class Σ^1_1 of *effective analytic sets* is the class of sets which are obtained by projection of arithmetical sets. It is well known that a set $L \subseteq \Sigma^\omega$, where Σ is a finite alphabet, is in the class Σ^1_1 iff it is accepted by a Turing machine with a Büchi or Muller acceptance condition [Sta97].
As indicated in the introduction, we made a mistake in the conference paper [Fin05]. We wrote there that it is well known that if $L \subseteq \Sigma^\omega$ is a (lightface) Σ^1_1 set, and is a Borel set of rank α, then α is smaller than ω^{CK}_1. This fact, which is true if we replace the (lightface) class Σ^1_1 by the (lightface) class Δ^1_1, is actually not true. Kechris, Marker and Sami proved in [KMS89] that the supremum of the set of Borel ranks of (lightface) Π^1_1, so also of (lightface) Σ^1_1, sets is the ordinal γ^1_2.
This ordinal is precisely defined in [KMS89]. Kechris, Marker and Sami proved that the ordinal γ^1_2 is strictly greater than the ordinal δ^1_2 which is the first non Δ^1_2 ordinal. Thus in particular it holds that $\omega^{\mathrm{CK}}_1 < \gamma^1_2$. The exact value of the

ordinal γ_2^1 may depend on axioms of set theory [KMS89]. It is consistent with the axiomatic system **ZFC** that γ_2^1 is equal to the ordinal δ_3^1 which is the first non Δ_3^1 ordinal (because $\gamma_2^1 = \delta_3^1$ in **ZFC + (V=L)**). On the other hand the axiom of Π_1^1-determinacy implies that $\gamma_2^1 < \delta_3^1$. For more details, the reader is referred to [KMS89] and to a textbook of set theory like [Jec02].

Notice however that it seems still unknown whether *every* non null ordinal $\gamma < \gamma_2^1$ is the Borel rank of a (lightface) Π_1^1 (or Σ_1^1) set. On the other hand it is known that every ordinal $\gamma < \omega_1^{\mathrm{CK}}$ is the Borel rank of a (lightface) Δ_1^1 set. Moreover, for every non null ordinal $\alpha < \omega_1^{\mathrm{CK}}$, there exist some $\mathbf{\Sigma}_\alpha^0$-complete and some $\mathbf{\Pi}_\alpha^0$-complete sets in the class Δ_1^1. Louveau gives the following argument: the natural universal set for the class $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$), where $\alpha < \omega_1^{\mathrm{CK}}$, is a $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete) set and it is in the class Δ_1^1, [Lou05]. The definition and the construction of a universal set for a given Borel class may be found in [Mos80].

4   Borel hierarchy of ω-languages in r-BCL(8)$_\omega$

It is well known that every Turing machine can be simulated by a (non real time) 2-counter automaton, see [HU79]. Thus the Borel hierarchy of the class $\mathbf{BCL}(2)_\omega$ is also the Borel hierarchy of the class of ω-languages accepted by Büchi Turing machines. We shall prove the following proposition.

Proposition 5. *The Borel hierarchy of the class* $\mathbf{r\text{-}BCL}(8)_\omega$ *is equal to the Borel hierarchy of the class* $\mathbf{BCL}(2)_\omega$.

We first sketch the proof of this result. We are going to find, from an ω-language $L \subseteq \Sigma^\omega$ in $\mathbf{BCL}(2)_\omega$, another ω-language $\theta_S(L)$ which will be of the same Borel complexity but accepted by a *real-time* 8-counter Büchi automaton. The idea is to add firstly a storage type called a queue to a 2-counter Büchi automaton in order to read ω-words in real-time. Then we shall see that a queue can be simulated by two pushdown stacks or by four counters. This simulation is not done in real-time but a crucial fact is that we can bound the number of transitions needed to simulate the queue. This allows to pad the strings in L with enough extra letters so that the new words will be read in real-time by a 8-counter Büchi automaton (two counters are used to check that an ω-word is really obtained with the good padding which is made in a regular way). The padding is obtained via the function θ_S which we define now.

Let Σ be an alphabet having at least two letters, E be a new letter not in Σ, S be an integer ≥ 1, and $\theta_S : \Sigma^\omega \to (\Sigma \cup \{E\})^\omega$ be the function defined, for all $x \in \Sigma^\omega$, by:

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\ldots$$

We now state the two following lemmas.

Lemma 6. *Let Σ be an alphabet having at least two letters and let $L \subseteq \Sigma^\omega$ be a subset of Σ^ω which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α) for some ordinal $\alpha \geq 2$. Then the ω-language $\theta_S(L)$ is a subset of $(\Sigma \cup \{E\})^\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).*

Proof. Let Σ be an alphabet having at least two letters. It is easy to see that the function θ_S is continuous because if two ω-words x and y of Σ^ω have a common initial segment of length n then the two ω-words $\theta_S(x)$ and $\theta_S(y)$ have a common initial segment of length greater than n.

Let $\varphi : (\Sigma \cup \{E\})^\omega \to (\Sigma \cup \{E\})^\omega$ be the mapping defined, for all $y \in (\Sigma \cup \{E\})^\omega$, by: $\varphi(y) = y(1).y(S+2).y(S+S^2+3)\ldots y(S+S^2+\ldots+S^n+(n+1))\ldots$

It is easy to see that the function φ is also continuous and that, for any $L \subseteq \Sigma^\omega$, $\theta_S(L) = \varphi^{-1}(L) \cap \theta_S(\Sigma^\omega)$.

Let now $L \subseteq \Sigma^\omega$ be a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of Σ^ω, hence also of $(\Sigma \cup \{E\})^\omega$, for some ordinal $\alpha \geq 2$. Then $\varphi^{-1}(L)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$ because the class $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) is closed under inverse images by continuous functions. On the other hand $\theta_S(\Sigma^\omega)$ is a closed set thus $\theta_S(L) = \varphi^{-1}(L) \cap \theta_S(\Sigma^\omega)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$ because the class $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) is closed under finite intersection.

Moreover it holds that $L = \theta_S^{-1}(\theta_S(L))$. Thus if L is assumed to be $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete) then $\theta_S(L)$ is a $\mathbf{\Sigma}^0_\alpha$-**complete** (respectively, $\mathbf{\Pi}^0_\alpha$-**complete**) subset of $(\Sigma \cup \{E\})^\omega$ because it is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) set, the function θ_S is continuous, and L is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete).

Assume now that L is a $\mathbf{\Sigma}^0_\alpha$-set of rank α, (respectively, $\mathbf{\Pi}^0_\alpha$-set of rank α). We have already seen that $\theta_S(L)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$. The Borel rank of $\theta_S(L)$ cannot be smaller than α because otherwise $L = \theta_S^{-1}(\theta_S(L))$ would be also of Borel rank smaller than α.

$\square$

Lemma 7. *Let Σ be an alphabet having at least two letters and let $L \subseteq \Sigma^\omega$ be an ω-language in the class $\mathbf{BCL}(2)_\omega$. Then there exists an integer $S \geq 1$ such that $\theta_S(L)$ is in the class $\mathbf{r}\text{-}\mathbf{BCL}(8)_\omega$.*

Proof. Let Σ be an alphabet having at least two letters and let $L \subseteq \Sigma^\omega$ be an ω-language accepted by a Büchi 2-counter automaton $\mathcal{A}$.

A way to construct a finite machine accepting the same ω-language but being *real time* would be to add a storage type called a *queue* [EH93].

Configurations of a queue are finite words over a finite alphabet Σ; a letter of Σ may be added to the rear of the queue or removed from the front; moreover there are tests to determine the first letter of the queue.

The new machine will read words in real time. At every transition a letter of the input ω-word is read, is added to the rear of the queue, waiting to be read (and

then removed from the front of the queue) for the simulation of the reading of the input ω-word by the 2-counter automaton $\mathcal{A}$.

We are going to see that one can simulate a queue with four counters. This simulation will not be a *real time* simulation but we shall be able to get an upper bound on the number of transitions of the four counters which are necessary for the simulation of one transition of the queue. This upper bound will be useful in the sequel for our purpose.

Claim 8. *A queue can be simulated by two pushdown stacks.*

Proof. Assume that the queue alphabet is $\Sigma = \{Z_2, Z_3, \ldots, Z_{k-1}\}$, for some integer $k \geq 3$.

The content of the queue can be represented by a finite word $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$, where the letter Z_{i_m} is the first letter of the queue and Z_{i_1} is the last letter of the queue (the last added to the rear).

This content can be stored in a pushdown stack whose alphabet is $\Gamma = \Sigma \cup \{Z_1\}$, where Z_1 is the bottom symbol which appears only at the bottom of the stack and always remains there. The stack content representing the queue content $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ will be simply $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.

If the letter Z_{i_m} of the front of the queue is removed from the queue, it suffices to pop the same letter from the top of the stack.

To simulate the addition of a new letter Z_r to the rear of the queue we can use a second pushdown stack whose alphabet is also Γ.

we have in fact to add the letter Z_r between the letters Z_1 and Z_{i_1} of the first pushdown stack. To achieve this, we successively pop letters from the top of the first stack, pushing them in the second stack containing only at the beginning the bottom symbol Z_1. After having done this operation for letters $Z_{i_m}, Z_{i_{m-1}}, \ldots Z_{i_2}, Z_{i_1}$, the content of the first stack is Z_1 and the content of the second stack is $Z_1 Z_{i_m} Z_{i_{m-1}} \ldots Z_{i_2} Z_{i_1}$. We can then push the letter Z_r at the top of the first stack. Then we successively pop letters $Z_{i_1}, Z_{i_2}, Z_{i_3} \ldots, Z_{i_m}$ from the top of the second stack, pushing them in the first stack. At the end of this operation the content of the first stack is $Z_0 Z_r Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ and it represents the new content of the queue. $\square$

We recall now the following well known property.

Claim 9 ([HU79]). *A pushdown stack can be simulated by two counters.*

Proof. Consider a stack having $k-1$ symbols $Z_1, Z_2, \ldots, Z_{k-1}$. The stack content $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ can be represented by the integer j which is given in base k by:

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \ldots + k^{m-1}.i_1$$

Notice that, as remarked in [HU79], not every integer represents a stack content. In particular an integer whose representation in base k contains the digit 0 does not represent any stack content.

We are going to see how to use a second counter to determine which is the letter at the top of the stack, and to simulate the operations of pushing a letter in the stack or of popping a letter from the top of the stack.

Assume that the integer j representing the stack content $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ is stored in one of the two counters.

In order to determine which is the letter at the top of the stack, we can copy the content j in the second counter, using the finite control of the finite machine to compute j modulo k.
The integer j modulo k is equal to the integer i_m which characterizes the letter Z_{i_m} hence the letter at the top of the stack.
It would be possible to transfer again the integer j in the first counter, but one can also leave it in the second counter and use the finite control to know in which counter is stored the integer j.
Notice that this operation needs only j steps (and $2j$ steps if we transfer again j in the first counter).

If a letter Z_r is pushed into the stack, the content of the stack is now $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m} Z_r$ and the integer associated with that content is $j.k + r$.
It is easy to store the integer $j.k$ in the second counter, by adding k to this counter each time the first counter is decreased by 1. When the content of the first counter is equal to zero then the content of the second counter is equal to $j.k$. One can then add r to the second counter by using the finite control of the machine. Again we can use the finite control to know that now the content of the stack is coded by the integer which is in the *second* counter.
Notice that the whole operation needs only $j.k + r$ steps.

If instead the symbol Z_{i_m} is popped from the top of the stack, then the new content of the stack is $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_{m-1}}$ and it is represented by the integer $[\frac{j}{k}] = i_{m-1} + k.i_{m-2} + \ldots + k^{m-2}.i_1$ which is the integer part of $\frac{j}{k}$.
To get the integer $[\frac{j}{k}]$ as content of the second counter, we can decrease the first counter from j to zero, adding 1 to the second counter each time the first one is decreased by k.
Notice that this operation needs only j steps.
Remark also that we can achieve this operation in a non deterministic way, checking at the end of it which was the letter at the top of the stack. □

We have seen above that a queue can be simulated with two pushdown stacks hence also with four counters.
This simulation is not done in *real time* but we shall see that we can get an upper bound on the number of transitions of the four counters simulating one transition of the queue. This upper bound will be crucial in view of Lemma 7.

Claim 10. *Assume as above that the queue alphabet is $\Sigma = \{Z_2, Z_3, \ldots, Z_{k-1}\}$, for some integer $k \geq 3$ and that at some time the content of the queue is a finite word $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$. Then the number of transitions of four counters which are needed to simulate the addition of a letter Z_r to the rear of the queue is smaller than $(2.k)^{m+2}$.*

Proof. Recall that the content of the queue can be stored in a pushdown stack whose alphabet is $\Gamma = \Sigma \cup \{Z_1\}$, where Z_1 is the bottom symbol. The stack content representing the queue content $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ is simply $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.

This stack content $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ can itself be represented by the integer j which is given in base k by:

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \ldots + k^{m-1}.i_1 + k^m.1 \ \leq \ k^{m+1}$$

We have seen that, considering the simulation of the addition of Z_r to the rear of a queue with two pushdown stacks, the first stack containing $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$, we have first to successively pop letters $Z_{i_m}, \ldots, Z_{i_3}, Z_{i_2}, Z_{i_1}$, from the top of the first stack and push them in the second stack.

We have also seen above that popping a letter from the stack whose content is represented by the integer j needs only j transitions of two counters and that we can know at the end of this popping simulation which letter has just been popped.

Moreover to push a letter Z_s in the stack whose content is represented by an integer j' needs only $j'.k + s$ transitions of two counters.

Thus at most $2.m.k^{m+1}$ transitions of four counters are necessary to simulate the operation of successively popping letters $Z_{i_m}, \ldots, Z_{i_3}, Z_{i_2}, Z_{i_1}$, from the top of the first stack and then pushing them in the second stack.

Two transitions of the counters are needed to check that the content of the first stack is now reduced to Z_1, which is simply represented by the integer 1, without changing this content; in one step the counter is reduced from one to zero then in a second step the counter is increased from zero to one.

To simulate the addition of the letter Z_r to the rear of the queue, we now push the letter Z_r in the first stack; this is simulated by $k + r$ transitions of two counters.

Now we have to successively pop letters $Z_{i_1}, Z_{i_2}, Z_{i_3}, \ldots, Z_{i_m}$, from the top of the second stack and push them again in the first stack. This whole operation needs less than $m.k^{m+1} + m.k^{m+2}$ transitions of the four counters.

Finally, to simulate the addition of a letter Z_r to the rear of the queue, we need only
$$2.m.k^{m+1} + 2 + k + r + m.k^{m+1} + m.k^{m+2}$$

transitions of four counters. This number is smaller than

$$4.m.k^{m+2} + 3.k \leq (2.k)^{m+2}.$$

$\square$

Claim 11. *Assume as above that the queue alphabet is $\Sigma = \{Z_2, Z_3, \ldots, Z_{k-1}\}$, for some integer $k \geq 3$ and that at some time the content of the queue is a finite word $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$. Then the number of transitions of four counters which are needed to determine the letter Z_{i_m} which is at the front of the queue is smaller than k^{m+1}. And the number of transitions of four counters which are needed to simulate the operation of removing the letter Z_{i_m} from the front of the queue is smaller than k^{m+1}.*

Proof. The content $Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$ of the queue is represented by a stack content $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \ldots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.
This stack content is itself represented by the integer

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \ldots + k^{m-1}.i_1 + k^m.1 \ \leq \ k^{m+1}$$

By the proof of Claim 9, $j \leq k^{m+1}$ transitions of four counters (and even of only two counters) suffice to determine the letter Z_{i_m} which is at the top of the stack or to pop it from the top of the stack. $\qquad \square$

Claim 12. *The ω-language $\theta_S(\Sigma^\omega)$ is in the class $\mathbf{r\text{-}BCL}(2)_\omega$.*

Proof. Recall that if Σ is an alphabet having at least two letters, E is a new letter not in Σ, S is an integer ≥ 1, then an ω-word $y \in (\Sigma \cup \{E\})^\omega$ is in $\theta_S(\Sigma^\omega)$ iff it is in the form:

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4) \ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}} \ldots$$

for some $x \in \Sigma^\omega$.
It is easy to construct a real time Büchi 2-counter automaton $\mathcal{B}$ accepting $\theta_S(\Sigma^\omega)$. We describe now the behaviour of $\mathcal{B}$ when reading an ω-word $y \in (\Sigma \cup \{E\})^\omega$. After the reading of the first letter $y(1) \in \Sigma$, the automaton $\mathcal{B}$ adds one to the first counter for each letter E read, checking with the finite control that there are S letters E following $y(1)$. Then $\mathcal{B}$ reads a second letter of Σ and next it adds S to the second counter and decreases the first counter by one each time it reads S letters E. When the first counter content is equal to zero, the second counter content is equal to S^2 and $\mathcal{B}$ has read S^2 letters E. It then reads a third letter of Σ, and next it adds S to the first counter and decreases the second counter by one each time it reads S letters E. When the second counter content is equal to zero, the first counter content is equal to S^3 and $\mathcal{B}$ has read S^3 letters E. Then $\mathcal{B}$ reads a fourth letter of Σ, and so on. The Büchi acceptance condition is used to check that the content of the first (and also of the second) counter takes infinitely many times the value zero. $\qquad \square$

    End of Proof of Lemma 7. Let $L \subseteq \Sigma^\omega$ be an ω-language accepted by a Büchi 2-counter automaton $\mathcal{A}$. We are going to explain the behaviour of a real time Büchi 8-counter automaton $\mathcal{A}_1$ accepting $\theta_S(L)$ where $S = (3k)^3$ and $k = cardinal(\Sigma) + 2$.
As explained in the proof of preceding Claim 12, two counters of $\mathcal{A}_1$ will be

used, independently of the other six ones, to check that the input ω-word $y \in (\Sigma \cup \{E\})^{\omega}$ is in $\theta_S(\Sigma^{\omega})$.

Consider now the reading by $\mathcal{A}_1$ of an ω-word $y \in (\Sigma \cup \{E\})^{\omega}$ in the form $y = \theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\ldots$ for some $x \in \Sigma^{\omega}$.

The automaton $\mathcal{A}_1$ will simulate, using four counters, a queue in which will be successively stored letters $x(1), x(2), \ldots, x(n), \ldots$ as soon as they will be read. Two other counters of $\mathcal{A}_1$ will be used to simulate the reading of the ω-word x by the Büchi 2-counter automaton $\mathcal{A}$. Notice that only letters $x(1), x(2), \ldots, x(n), \ldots$ will be added to the rear of the queue. Therefore after having read the initial segment $x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}$ of y the content of the queue has cardinal smaller than or equal to n.

When $\mathcal{A}_1$ reads $x(n+1)$ it will firstly simulate the addition of the letter $x(n+1)$ to the rear of the queue, using four counters, and doing this in real time while continuing to read some following letters E. By Claim 10, the number of transitions of four counters needed to simulate the addition of $x(n+1)$ to the rear of the queue is smaller than $(2k)^{n+2}$ where $k = cardinal(\Sigma) + 2$. Next $\mathcal{A}_1$ determines which is the letter at the front of the queue. By Claim 11 this needs at most k^{n+2} transitions of the four counters (because there are now at most $(n+1)$ letters in the queue). Now the automaton $\mathcal{A}_1$ simulates, using two counters, only one transition of $\mathcal{A}$. This transition may be a λ-transition or not. In the second case $\mathcal{A}_1$ simulates the reading by $\mathcal{A}$ of the letter at the front of the queue so this letter is removed from the queue; by Claim 11 this needs again at most k^{n+2} transitions of the four counters. It holds that

$$(2k)^{n+2} + k^{n+2} + k^{n+2} + 1 \leq ((3k)^3)^n$$

Thus if we set $S = (3k)^3$ then all these transitions of the six counters can be achieved by the automaton $\mathcal{A}_1$ in *real-time* during the reading of the letters E following $x(n+1)$ in y. Not all the S^{n+1} letters E are read during these transitions of the six counters. But $\mathcal{A}_1$ will read the other ones without changing the contents of the six counters, waiting for the reading of the next letter of Σ: the letter $x(n+2)$. It will then simulate the addition of this letter to the rear of the queue, and so on.

A Muller condition can be used to ensure that $y \in \theta_S(\Sigma^{\omega})$, i.e. $y = \theta_S(x)$ for some $x \in \Sigma^{\omega}$, *and* that $x \in L = L(\mathcal{A})$. As mentioned in Section 2, this can also be achieved with a Büchi acceptance condition. $\qquad\square$

Notice that we did not aim to find the smallest possible integer S but only to find one such integer in order to prove Lemma 7.

5 Borel hierarchy of ω-languages in r-BCL(1)$_{\omega}$

We shall firstly prove the following result.

Proposition 13. *Let $k \geq 2$ be an integer. If, for some ordinal $\alpha \geq 2$, there is an ω-language in the class $\mathbf{r}\text{-}\mathbf{BCL}(k)_\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α), then there is some ω-language in the class $\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).*

To simplify the exposition of the proof of this result, firstly, we are going to give the proof for $k = 2$. Next we shall explain the modifications to do in order to infer the result for the integer $k = 8$ which is in fact the only case we shall need in the sequel. (However our main result will show that the proposition is true for every integer $k \geq 2$).
For that purpose we define first a coding of ω-words over a finite alphabet Σ by ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ where A, B and 0 are new letters not in Σ. We shall code an ω-word $x \in \Sigma^\omega$ by the ω-word $h(x)$ defined by

$$h(x) = A.0^6.x(1).B.0^{6^2}.A.0^{6^2}.x(2).B.0^{6^3}.A.0^{6^3}.x(3).B \ldots B.0^{6^n}.A.0^{6^n}.x(n).B \ldots$$

This coding defines a mapping $h : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$. The function h is continuous because for all ω-words $x, y \in \Sigma^\omega$ and each positive integer n, it holds that $\delta(x, y) < 2^{-n} \to \delta(h(x), h(y)) < 2^{-n}$.

Lemma 14. *Let Σ be a finite alphabet and $(h(\Sigma^\omega))^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$. If $\mathcal{L} \subseteq \Sigma^\omega$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α), for a countable ordinal $\alpha \geq 2$, then $h(\mathcal{L}) \cup h(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).*

Proof. The topological space Σ^ω is compact thus its image by the continuous function h is also a compact subset of the topological space $(\Sigma \cup \{A, B, 0\})^\omega$. The set $h(\Sigma^\omega)$ is compact hence it is a closed subset of $(\Sigma \cup \{A, B, 0\})^\omega$. Then its complement

$$(h(\Sigma^\omega))^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$$

is an open (i.e. a $\mathbf{\Sigma}^0_1$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$.

On the other hand the function h is also injective thus it is a bijection from Σ^ω onto $h(\Sigma^\omega)$. But a continuous bijection between two compact sets is an homeomorphism therefore h induces an homeomorphism between Σ^ω and $h(\Sigma^\omega)$. Assume that $\mathcal{L}$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of Σ^ω. Then $h(\mathcal{L})$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $h(\Sigma^\omega)$ (where Borel sets of the topological space $h(\Sigma^\omega)$ are defined from open sets as in the case of the topological space Σ^ω).

The topological space $h(\Sigma^\omega)$ is a topological subspace of $(\Sigma \cup \{A, B, 0\})^\omega$ and its topology is induced by the topology on $(\Sigma \cup \{A, B, 0\})^\omega$: open sets of $h(\Sigma^\omega)$ are traces on $h(\Sigma^\omega)$ of open sets of $(\Sigma \cup \{A, B, 0\})^\omega$ and the same result holds for closed sets. Then one can easily show by induction that for every ordinal

$\alpha \geq 1$, $\mathbf{\Pi}_\alpha^0$-subsets (resp. $\mathbf{\Sigma}_\alpha^0$-subsets) of $h(\Sigma^\omega)$ are traces on $h(\Sigma^\omega)$ of $\mathbf{\Pi}_\alpha^0$-subsets (resp. $\mathbf{\Sigma}_\alpha^0$-subsets) of $(\Sigma \cup \{A, B, 0\})^\omega$, i.e. are intersections with $h(\Sigma^\omega)$ of $\mathbf{\Pi}_\alpha^0$-subsets (resp. $\mathbf{\Sigma}_\alpha^0$-subsets) of $(\Sigma \cup \{A, B, 0\})^\omega$.

But $h(\mathcal{L})$ is a $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$)-subset of $h(\Sigma^\omega)$, for some ordinal $\alpha \geq 2$, hence there exists a $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subset T of $(\Sigma \cup \{A, B, 0\})^\omega$ such that $h(\mathcal{L}) = T \cap h(\Sigma^\omega)$. But $h(\Sigma^\omega)$ is a closed i.e. $\mathbf{\Pi}_1^0$-subset of $(\Sigma \cup \{A, B, 0\})^\omega$ and the class of $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subsets of $(\Sigma \cup \{A, B, 0\})^\omega$ is closed under finite intersection thus $h(\mathcal{L})$ is a $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$.

Now $h(\mathcal{L}) \cup (h(\Sigma^\omega))^-$ is the union of a $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subset and of a $\mathbf{\Sigma}_1^0$-subset of $(\Sigma \cup \{A, B, 0\})^\omega$ therefore it is a $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$ because the class of $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$) subsets of $(\Sigma \cup \{A, B, 0\})^\omega$ is closed under finite union.

Assume now firstly that $\mathcal{L}$ is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete). In order to prove that $h(\mathcal{L}) \cup (h(\Sigma^\omega))^-$ is $\mathbf{\Sigma}_\alpha^0$-**complete** (respectively, $\mathbf{\Pi}_\alpha^0$-**complete**) it suffices to remark that

$$\mathcal{L} = h^{-1}[h(\mathcal{L}) \cup (h(\Sigma^\omega))^-]$$

This implies that $h(\mathcal{L}) \cup (h(\Sigma^\omega))^-$ is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete) because $\mathcal{L}$ is assumed to be $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete).

On the other hand if we assume only that $\mathcal{L}$ is a $\mathbf{\Sigma}_\alpha^0$-set of rank α (respectively, $\mathbf{\Pi}_\alpha^0$-set of rank α), then we can infer that $h(\mathcal{L}) \cup (h(\Sigma^\omega))^-$ is also a $\mathbf{\Sigma}_\alpha^0$-set of rank α (respectively, $\mathbf{\Pi}_\alpha^0$-set of rank α). Indeed if $h(\mathcal{L}) \cup (h(\Sigma^\omega))^-$ was of Borel rank $\beta < \alpha$ then $\mathcal{L} = h^{-1}[h(\mathcal{L}) \cup (h(\Sigma^\omega))^-]$ would be also of rank smaller than α because the class $\mathbf{\Sigma}_\beta^0$ (respectively, $\mathbf{\Pi}_\beta^0$) is closed under inverse images by continuous functions. $\qquad\square$

In order to apply Lemma 14, we want now to prove that if $L(\mathcal{A}) \subseteq \Sigma^\omega$ is accepted by a real time 2-counter automaton $\mathcal{A}$ with a Büchi acceptance condition then $h(L(\mathcal{A})) \cup h(\Sigma^\omega)^-$ is accepted by a 1-counter automaton with a Büchi acceptance condition. We firstly prove the following lemma.

Lemma 15. *Let Σ be a finite alphabet and h be the coding of ω-words over Σ defined as above. Then $h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$ is accepted by a real time 1-counter Büchi automaton.*

Proof. We can easily see that $h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$ is the set of ω-words in $(\Sigma \cup \{A, B, 0\})^\omega$ which belong to one of the following ω-languages.

- $\mathcal{D}_1$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ which have not any initial segment in $A.0^6.\Sigma.B$. It is easy to see that $\mathcal{D}_1$ is in fact a regular ω-language.

– $\mathcal{D}_2$ is the complement of $(A.0^+.\Sigma.B.0^+)^\omega$ in $(\Sigma\cup\{A,B,0\})^\omega$. The ω-language $(A.0^+.\Sigma.B.0^+)^\omega$ is regular thus its complement $\mathcal{D}_2$ is also a regular ω-language.

– $\mathcal{D}_3$ is the set of ω-words over the alphabet $\Sigma\cup\{A,B,0\}$ which contain a segment in $B.0^n.A.0^m.\Sigma$ for some positive integers $n\neq m$. It is easy to see that this ω-language can be accepted by a real time 1-counter Büchi automaton.

– $\mathcal{D}_4$ is the set of ω-words over the alphabet $\Sigma\cup\{A,B,0\}$ which contain a segment in $A.0^n.\Sigma.B.0^m.A$ for some positive integers n and m with $m\neq 6n$. Again this ω-language can be accepted by a real time 1-counter Büchi automaton.

The class $\mathbf{r\text{-}BCL}(1)_\omega$ is closed under finite union because it is the class of ω-languages accepted by *non deterministic* real time 1-counter Büchi automata. On the other hand it holds that $h(\Sigma^\omega)^- = (\Sigma\cup\{A,B,0\})^\omega - h(\Sigma^\omega) = \cup_{1\leq i\leq 4}\mathcal{D}_i$ thus $h(\Sigma^\omega)^-$ is accepted by a real time 1-counter Büchi automaton. $\square$

We would like now to prove that if $L(\mathcal{A})\subseteq \Sigma^\omega$ is accepted by a real time 2-counter automaton $\mathcal{A}$ with a Büchi acceptance condition then $h(L(\mathcal{A}))$ is in $\mathbf{BCL}(1)_\omega$. We cannot show this, so we are firstly going to define another ω-language $\mathcal{L}(\mathcal{A})$ accepted by a 1-counter Büchi automaton and we shall prove that $h(L(\mathcal{A}))\cup h(\Sigma^\omega)^- = \mathcal{L}(\mathcal{A})\cup h(\Sigma^\omega)^-$.

We shall need the following notion. Let $N\geq 1$ be an integer such that $N = 2^x.3^y.N_1$ where x,y are positive integers and $N_1\geq 1$ is an integer which is neither divisible by 2 nor by 3. Then we set $P_2(N) = x$ and $P_3(N) = y$. So $2^{P_2(N)}$ is the greatest power of 2 which divides N and $2^{P_3(N)}$ is the greatest power of 3 which divides N.

Let then a 2-counter Büchi automaton $\mathcal{A} = (K,\Sigma,\Delta,q_0,F)$ accepting the ω-language $L(\mathcal{A})\subseteq \Sigma^\omega$. The ω-language $\mathcal{L}(\mathcal{A})$ is the set of ω-words over the alphabet $\Sigma\cup\{A,B,0\}$ in the form

$$A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A\ldots A.u_n.v_n.x_n.B.w_n.z_n.A\ldots$$

where, for all integers $i\geq 1$, $v_i,w_i\in 0^+$, $u_i,z_i\in 0^\star$, $x_i\in\Sigma$, $|u_1|=5$, $|u_{i+1}|=|z_i|$ and there is a sequence $(q_i)_{i\geq 0}$ of states of K and integers $j_i,j_i'\in\{-1;0;1\}$, for $i\geq 1$, such that for all integers $i\geq 1$:

$$x_i : (q_{i-1},P_2(|v_i|),P_3(|v_i|)) \mapsto_{\mathcal{A}} (q_i,P_2(|v_i|)+j_i,P_3(|v_i|)+j_i')$$

and

$$|w_i| = |v_i|.2^{j_i}.3^{j_i'}$$

Moreover some state $q_f\in F$ occurs infinitely often in the sequence $(q_i)_{i\geq 0}$. Notice that the state q_0 of the sequence $(q_i)_{i\geq 0}$ is also the initial state of $\mathcal{A}$.

Lemma 16. *Let $\mathcal{A}$ be a real time 2-counter Büchi automaton accepting ω-words over the alphabet Σ and $\mathcal{L}(\mathcal{A})\subseteq (\Sigma\cup\{A,B,0\})^\omega$ be defined as above. Then $\mathcal{L}(\mathcal{A})$ is accepted by a 1-counter Büchi automaton $\mathcal{B}$.*

Proof. We shall explain informally the behaviour of a 1-counter Büchi automaton $\mathcal{B}$ accepting the ω-language $\mathcal{L}(\mathcal{A})$.

We firstly consider the reading of an ω-word $x \in (A.0^\star.\Sigma.B.0^\star)^\omega$ in the form

$$x = A.0^{n_1} x_1.B.0^{m_1}.A.0^{n_2} x_2.B.0^{m_2}.A \ldots A.0^{n_p} x_p.B.0^{m_p}.A \ldots$$

where, for all integers $i \geq 1$, n_i, m_i, are positive integers and $x_i \in \Sigma$.

Using the finite control the automaton $\mathcal{B}$ first checks that the six first letters of x form the initial segment $A.0^5$. Then, when reading the following $(n_1 - 5)$ letters 0, the automaton $\mathcal{B}$, using the finite control, checks that $(n_1 - 5) > 0$ and determines whether $P_2(n_1 - 5) = 0$ and whether $P_3(n_1 - 5) = 0$. Moreover the counter content is increased by one for each letter 0 read. The automaton $\mathcal{B}$ reads now the letter x_1 and it guesses a transition of $\mathcal{A}$ leading to

$$x_1 : (q_0, P_2(n_1 - 5), P_3(n_1 - 5)) \mapsto_\mathcal{A} (q_1, P_2(n_1 - 5) + j_1, P_3(n_1 - 5) + j_1')$$

We set $v_1 = 0^{n_1-5}$ and $w_1 = 0^{(n_1-5).2^{j_1}.3^{j_1'}}$. The counter value is now equal to $(n_1 - 5)$ and, when reading letters 0 following x_1, the automaton $\mathcal{B}$ checks that $m_1 \geq (n_1 - 5).2^{j_1}.3^{j_1'}$ in such a way that the counter value becomes 0 after having read the $(n_1 - 5).2^{j_1}.3^{j_1'}$ letters 0 following the first letter B. For instance if $j_1 = j_1' = 1$ then $|w_1| = |v_1|.6$ so this can be done by decreasing the counter content by one each time six letters 0 are read. The other cases are treated in a similar way. Details are here left to the reader.

Notice also that the automaton $\mathcal{B}$ has kept in its finite control the value of the state q_1.

We set now $0^{m_1} = w_1.z_1$. We have seen that after having read w_1 the counter value is equal to zero. Now when reading z_1 the counter content is increased by one for each letter read so that it becomes $|z_1|$ after having read z_1. The automaton $\mathcal{B}$ reads now a letter A and next decreased its counter by one for each letter 0 read until the counter content is equal to zero. We set $0^{n_2} = u_2.v_2$ with $u_2 = z_1$. The automaton $\mathcal{B}$ reads now the segment v_2. Using the finite control, it checks that $|v_2| > 0$ and determines whether $P_2(|v_2|) = 0$ and whether $P_3(|v_2|) = 0$. Moreover the counter content is increased by one for each letter 0 read. The automaton $\mathcal{B}$ reads now the letter x_2 and it guesses a transition of $\mathcal{A}$ leading to

$$x_2 : (q_1, P_2(|v_2|), P_3(|v_2|)) \mapsto_\mathcal{A} (q_2, P_2(|v_2|) + j_2, P_3(|v_2|) + j_2')$$

We set $w_2 = 0^{|v_2|.2^{j_2}.3^{j_2'}}$. The counter value is now equal to $|v_2|$. The automaton $\mathcal{B}$ reads now the second letter B and, when reading the m_2 letters 0 following this letter B, the automaton $\mathcal{B}$ checks that $m_2 \geq |v_2|.2^{j_2}.3^{j_2'}$ in such a way that the counter value becomes 0 after having read the $|v_2|.2^{j_2}.3^{j_2'}$ letters 0 following the second letter B.

For instance if $j_2 = 0$ and $j_2' = -1$ then $|w_2| = |v_2|.3^{-1}$ so this can be done by decreasing the counter content by three each time one letter 0 is read.

And if $j_2 = -1$ and $j_2' = -1$ then $|w_2| = |v_2|.2^{-1}.3^{-1} = |v_2|.6^{-1}$ so this can be done by decreasing the counter content by six each time one letter 0 is read. The

other cases are treated in a similar way. Details are here left to the reader.

Notice that these different cases can be achieved with the use of λ-transitions but in such a way that there will be at most 5 consecutive λ-transitions during a run of $\mathcal{B}$ on x. This will be an important useful fact in the sequel.

Notice also that the automaton $\mathcal{B}$ has kept in its finite control the value of the state q_2.

The reading of x by $\mathcal{B}$ continues in the same way. A Büchi acceptance condition can be used to ensure that some state $q_f \in K$ occurs infinitely often in the sequence $(q_i)_{i \geq 0}$.

To complete the proof we can remark that $\mathcal{R} = (A.0^\star.\Sigma.B.0^\star)^\omega$ is a regular ω-language so we have considered only the reading by $\mathcal{B}$ of ω-words $x \in \mathcal{R}$. Indeed if the ω-language $L(\mathcal{B})$ was not included into $\mathcal{R}$ we could replace it by $L(\mathcal{B}) \cap \mathcal{R}$ because the class $\mathbf{BCL}(1)_\omega$ is closed under intersection with regular ω-languages (by a classical construction of product of automata, the ω-language $\mathcal{R}$ being accepted by a deterministic Muller automaton). $\square$

Lemma 17. *Let $\mathcal{A}$ be a real time 2-counter Büchi automaton accepting ω-words over the alphabet Σ and $\mathcal{L}(\mathcal{A}) \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ be defined as above. Then $L(\mathcal{A})$ $= h^{-1}(\mathcal{L}(\mathcal{A}))$, i.e. $\forall x \in \Sigma^\omega \quad h(x) \in \mathcal{L}(\mathcal{A}) \longleftrightarrow x \in L(\mathcal{A})$.*

Proof. Let $\mathcal{A}$ be a real time 2-counter Büchi automaton accepting ω-words over the alphabet Σ and $\mathcal{L}(\mathcal{A}) \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ be defined as above. Let $x \in \Sigma^\omega$ be an ω-word such that $h(x) \in \mathcal{L}(\mathcal{A})$. So $h(x)$ may be written

$$h(x) = A.0^6.x(1).B.0^{6^2}.A.0^{6^2}.x(2).B.0^{6^3}.A.0^{6^3}.x(3).B \ldots B.0^{6^n}.A.0^{6^n}.x(n).B \ldots$$

and also

$$h(x) = A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A \ldots A.u_n.v_n.x_n.B.w_n.z_n.A \ldots$$

where, for all integers $i \geq 1$, $v_i, w_i \in 0^+$, $u_i, z_i \in 0^\star$, $x_i = x(i) \in \Sigma$, $|u_1| = 5$, $|u_{i+1}| = |z_i|$ and there is a sequence $(q_i)_{i \geq 0}$ of states of K and integers $j_i, j'_i \in \{-1; 0; 1\}$, for $i \geq 1$, such that for all integers $i \geq 1$:

$$x_i : (q_{i-1}, P_2(|v_i|), P_3(|v_i|)) \mapsto_{\mathcal{A}} (q_i, P_2(|v_i|) + j_i, P_3(|v_i|) + j'_i)$$

and

$$|w_i| = |v_i|.2^{j_i}.3^{j'_i}$$

some state $q_f \in F$ occurring infinitely often in the sequence $(q_i)_{i \geq 0}$.

In particular, $u_1 = 0^5$ and $u_1.v_1 = 0^6$ thus $|v_1| = 1 = 2^0.3^0$. We are going to prove by induction on the integer $i \geq 1$ that, for all integers $i \geq 1$, $|w_i| = |v_{i+1}| = 2^{P_2(|w_i|)}.3^{P_3(|w_i|)}$. Moreover, setting $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$, we are going to prove that for each integer $i \geq 1$ it holds that

$$x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathcal{A}} (q_i, c_1^{i+1}, c_2^{i+1})$$

We have already seen that $|v_1| = 1 = 2^0.3^0$. By hypothesis there is a state $q_1 \in K$ and integers $j_1, j'_1 \in \{-1; 0; 1\}$ such that $x_1 : (q_0, P_2(|v_1|), P_3(|v_1|)) \mapsto_{\mathcal{A}}$

$(q_1, P_2(|v_1|) + j_1, P_3(|v_1|) + j'_1)$, i.e. $x_1 : (q_0, 0, 0) \mapsto_{\mathcal{A}} (q_1, j_1, j'_1)$. Then $|w_1| = |v_1|.2^{j_1}.3^{j'_1} = 2^{j_1}.3^{j'_1}$.

We have now $|w_1.z_1| = |u_2.v_2| = 0^{6^2}$ and $|u_2| = |z_1|$ thus $|v_2| = |w_1| = 2^{j_1}.3^{j'_1}$. Setting $c_1^1 = 0$, $c_2^1 = 0$, $c_1^2 = j_1 = P_2(|v_2|)$ and $c_2^2 = j'_1 = P_3(|v_2|)$, it holds that $x_1 : (q_0, c_1^1, c_2^1) \mapsto_{\mathcal{A}} (q_1, c_1^2, c_2^2)$.

Assume now that, for all integers i, $1 \leq i \leq n - 1$, it holds that $|w_i| = |v_{i+1}| = 2^{P_2(|w_i|)}.3^{P_3(|w_i|)}$ and $x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathcal{A}} (q_i, c_1^{i+1}, c_2^{i+1})$ where $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$.

We know that there is a state $q_n \in K$ and integers $j_n, j'_n \in \{-1; 0; 1\}$ such that $x_n : (q_{n-1}, P_2(|v_n|), P_3(|v_n|)) \mapsto_{\mathcal{A}} (q_n, P_2(|v_n|) + j_n, P_3(|v_n|) + j'_n)$, i.e. $x_n : (q_{n-1}, c_1^n, c_2^n) \mapsto_{\mathcal{A}} (q_n, c_1^n + j_n, c_2^n + j'_n)$. Then $|w_n| = |v_n|.2^{j_n}.3^{j'_n} = 2^{c_1^n + j_n}.3^{c_2^n + j'_n}$. On the other hand $|w_n.z_n| = |u_{n+1}.v_{n+1}| = 0^{6^{n+1}}$ and $|u_{n+1}| = |z_n|$ thus $|v_{n+1}| = |w_n| = 2^{c_1^n + j_n}.3^{c_2^n + j'_n} = 2^{c_1^{n+1}}.3^{c_2^{n+1}}$ by setting $c_1^{n+1} = P_2(|v_{n+1}|)$ and $c_2^{n+1} = P_3(|v_{n+1}|)$. So we have $x_n : (q_{n-1}, c_1^n, c_2^n) \mapsto_{\mathcal{A}} (q_n, c_1^{n+1}, c_2^{n+1})$.

Finally we have proved by induction the announced claim. If for all integers $i \geq 1$, we set $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$, it holds that

$$x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathcal{A}} (q_i, c_1^{i+1}, c_2^{i+1})$$

But there is some state $q_f \in K$ which occurs infinitely often in the sequence $(q_i)_{i \geq 1}$. This implies that $(q_{i-1}, c_1^i, c_2^i)_{i \geq 1}$ is a successful run of $\mathcal{A}$ on x thus $x \in L(\mathcal{A})$.

Conversely it is easy to see that if $x \in L(\mathcal{A})$ then $h(x) \in \mathcal{L}(\mathcal{A})$. This ends the proof of Lemma 17. $\qquad\square$

Remark 18. *The simulation, during the reading of $h(x)$ by the 1-counter Büchi automaton $\mathcal{B}$, of the behaviour of the real time 2-counter Büchi automaton $\mathcal{A}$ reading x, can be achieved, using a coding of the content (c_1, c_2) of two counters by a single integer $2^{c_1}.3^{c_2}$ and the **special shape** of ω-words in $h(\Sigma^\omega)$ which allows the propagation of the counter value of $\mathcal{B}$. This will be sufficient here, because of the previous lemmas, and in particular of the fact that $h(\Sigma^\omega)^-$ is in the class $\mathbf{r\text{-}BCL}(1)_\omega$, and we can now end the proof of Proposition 13.*

End of Proof of Proposition 13. Let $\alpha \geq 2$ be an ordinal. Assume that there is an ω-language $L(\mathcal{A}) \subseteq \Sigma^\omega$ which is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α) and is accepted by a real time 2-counter Büchi automaton $\mathcal{A}$. By Lemma 14, $h(\mathcal{L}) \cup h(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ being $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α). On the other hand Lemma 17 states that $L(\mathcal{A}) = h^{-1}(\mathcal{L}(\mathcal{A}))$ and this implies that $h(L(\mathcal{A})) \cup h(\Sigma^\omega)^- = \mathcal{L}(\mathcal{A}) \cup h(\Sigma^\omega)^-$. But we know by Lemmas 15 and 16 that the ω-languages $h(\Sigma^\omega)^-$ and $\mathcal{L}(\mathcal{A})$ are in the class $\mathbf{BCL}(1)_\omega$ thus their union is also accepted by a 1-counter Büchi automaton. Therefore $h(L(\mathcal{A})) \cup h(\Sigma^\omega)^-$ is an ω-language in the class $\mathbf{BCL}(1)_\omega$ which is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α).

We want now to find an ω-language in the class $\mathbf{r\text{-}BCL}(1)_\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

On one side we have proved that $h(\Sigma^\omega)^-$ is accepted by a *real time* 1-counter Büchi automaton. On the other side we have proved that $\mathcal{L}(\mathcal{A})$ is accepted by a (non real time) 1-counter Büchi automaton $\mathcal{B}$. However we have seen, in the proof of Lemma 16, that at most 5 consecutive λ-transitions can occur during the reading of an ω-word x by $\mathcal{B}$.

Consider now the mapping $\phi : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ which is defined, for all $x \in (\Sigma \cup \{A, B, 0\})^\omega$ by:

$$\phi(x) = F^5.x(1).F^5.x(2).F^5.x(3)\ldots F^5.x(n).F^5.x(n+1).F^5 \ldots$$

The function ϕ is continuous and separates two successive letters of x by five letters F. We can prove, as in the proof of Lemma 6, that if $L \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α), for some ordinal $\alpha \geq 2$, then $\phi(L)$ is a subset of $(\Sigma \cup \{A, B, F, 0\})^\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

Thus the ω-language $\phi(\mathcal{L}(\mathcal{A}) \cup h(\Sigma^\omega)^-)$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

Moreover it is easy to see that $\phi(\mathcal{L}(\mathcal{A}))$ is accepted by a *real time* 1-counter Büchi automaton $\mathcal{B}'$. The automaton $\mathcal{B}'$ checks with its finite control that an input ω-word is in the form $\phi(x)$ for some $x \in (\Sigma \cup \{A, B, 0\})^\omega$. And $\mathcal{B}'$ simulates the reading of x by $\mathcal{B}$, the λ-transitions of $\mathcal{B}$ occuring during the reading, in *real time*, of letters F of the ω-word $\phi(x)$.

Finally $\phi(\mathcal{L}(\mathcal{A}) \cup h(\Sigma^\omega)^-) = \phi(\mathcal{L}(\mathcal{A})) \cup \phi(h(\Sigma^\omega)^-))$ is the union of two ω-languages in $\mathbf{r\text{-}BCL}(1)_\omega$ thus it is in $\mathbf{r\text{-}BCL}(1)_\omega$ and it is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

This ends the proof of Proposition 13 for the integer $k = 2$.

We explain now the modifications to do in order to prove Proposition 13 for the integer $k = 8$. We assume that $\alpha \geq 2$ is an ordinal and that there is an ω-language $L(\mathcal{A}) \subseteq \Sigma^\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α) and is accepted by a real time 8-counter Büchi automaton $\mathcal{A}$.

We first modify the coding of ω-words which was given by the mapping h. We replace the number $6 = 2.3$ by the product of the eight first prime numbers:

$$K = 2.3.5.7.11.13.17.19 = 9699690$$

Then an ω-word $x \in \Sigma^\omega$ will be coded by the ω-word

$$h_K(x) = A.0^K.x(1).B.0^{K^2}.A.0^{K^2}.x(2).B.0^{K^3}.A.0^{K^3}.x(3).B \ldots B.0^{K^n}.A.0^{K^n}.x(n).B \ldots$$

The mapping $h_K : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$ is continuous and we can prove, as in Lemma 14, that $h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ which is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α). As in

Lemma 15, we can prove that $h_K(\Sigma^\omega)^-$ is in the class **r-BCL**$(1)_\omega$.

Next, for each prime number $p \in \{2; 3; 5; 7; 11; 13; 17; 19\}$, and each positive integer $N \geq 1$, we denote $P_p(N)$ the positive integer such that $p^{P_p(N)}$ is the greatest power of p which divides N.

We define the ω-language $\mathcal{L}(\mathcal{A})$ as the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ in the form

$$A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A \ldots A.u_n.v_n.x_n.B.w_n.z_n.A \ldots$$

where, for all integers $i \geq 1$, $v_i, w_i \in 0^+$, $u_i, z_i \in 0^\star$, $|u_1| = K-1$, $|u_{i+1}| = |z_i|$ and there is a sequence $(q_i)_{i \geq 0}$ of states of K and integers $j_i^1, j_i^2, \ldots, j_i^8, \in \{-1; 0; 1\}$, for $i \geq 1$, such that for all integers $i \geq 1$:

$$x_i : (q_{i-1}, P_2(|v_i|), P_3(|v_i|), \ldots, P_{19}(|v_i|)) \mapsto_{\mathcal{A}} (q_i, P_2(|v_i|)+j_i^1, P_3(|v_i|)+j_i^2, \ldots, P_{19}(|v_i|)+j_i^8)$$

and

$$|w_i| = |v_i|.2^{j_i^1}.3^{j_i^2}.\ldots.19^{j_i^8}$$

and some state $q_f \in F$ occurs infinitely often in the sequence $(q_i)_{i \geq 0}$.

Applying the same ideas as in the proofs of Lemmas 16 and 17 we can prove that $\mathcal{L}(\mathcal{A})$ is accepted by a 1-counter Büchi automaton and that $L(\mathcal{A}) = h_K^{-1}(\mathcal{L}(\mathcal{A}))$. The essential change is that now the content $(c_1, c_2, \ldots, c_8)$ of eight counters is coded by the product $2^{c_1}.3^{c_2}.\ldots.(17)^{c_7}.(19)^{c_8}$.

Notice that again $\mathcal{L}(\mathcal{A})$ is accepted by a (non real time) 1-counter Büchi automaton $\mathcal{B}$. However there are now at most $(K-1)$ consecutive λ-transitions which can occur during the reading of an ω-word x by $\mathcal{B}$.

So we define now the mapping $\phi_K : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ by: for all $x \in (\Sigma \cup \{A, B, 0\})^\omega$,

$$\phi_K(x) = F^{K-1}.x(1).F^{K-1}.x(2).F^{K-1}.x(3) \ldots F^{K-1}.x(n).F^{K-1}.x(n+1).F^{K-1} \ldots$$

The function ϕ_K is continuous as the function ϕ was. The end of the proof is unchanged so we infer that $\phi_K(h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^-)$ is an ω-language in the class **r-BCL**$(1)_\omega$ which is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α). $\qquad\square$

From the results of Section 4 and Proposition 13, we can now state the following result.

Theorem 19. *Let $\mathcal{C}$ be a class of ω-languages such that:*

$$\textbf{r-BCL}(1)_\omega \subseteq \mathcal{C} \subseteq \Sigma_1^1.$$

(a) The Borel hierarchy of the class $\mathcal{C}$ is equal to the Borel hierarchy of the class Σ_1^1.

(b) $\gamma_2^1 = Sup \ \{\alpha \mid \exists L \in \mathcal{C}$ such that L is a Borel set of rank $\alpha\}$.

(c) For every non null ordinal $\alpha < \omega_1^{\text{CK}}$, there exists some $\mathbf{\Sigma}_\alpha^0$-complete and some $\mathbf{\Pi}_\alpha^0$-complete ω-languages in the class $\mathcal{C}$.

Notice that above (b) and (c) just follow from (a) and from the known results about the Borel hierarchy of the class Σ_1^1.

6 Wadge hierarchy of ω-languages in r-BCL(1)$_\omega$

We now introduce the Wadge hierarchy, which is a great refinement of the Borel hierarchy defined via reductions by continuous functions, [Dup01,Wad83].

Definition 20 (Wadge [Wad83]). *Let X, Y be two finite alphabets. For $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, L is said to be Wadge reducible to L' ($L \leq_W L'$) iff there exists a continuous function $f : X^\omega \to Y^\omega$, such that $L = f^{-1}(L')$.*
L and L' are Wadge equivalent iff $L \leq_W L'$ and $L' \leq_W L$. This will be denoted by $L \equiv_W L'$. And we shall say that $L <_W L'$ iff $L \leq_W L'$ but not $L' \leq_W L$.
A set $L \subseteq X^\omega$ is said to be self dual iff $L \equiv_W L^-$, and otherwise it is said to be non self dual.

The relation $\leq_W$ is reflexive and transitive, and $\equiv_W$ is an equivalence relation.
The *equivalence classes* of $\equiv_W$ are called *Wadge degrees*.
The Wadge hierarchy WH is the class of Borel subsets of a set X^ω, where X is a finite set, equipped with $\leq_W$ and with $\equiv_W$.
For $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, if $L \leq_W L'$ and $L = f^{-1}(L')$ where f is a continuous function from X^ω into Y^ω, then f is called a continuous reduction of L to L'.
Intuitively it means that L is less complicated than L' because to check whether $x \in L$ it suffices to check whether $f(x) \in L'$ where f is a continuous function.
Hence the Wadge degree of an ω-language is a measure of its topological complexity.
Notice that in the above definition, we consider that a subset $L \subseteq X^\omega$ is given together with the alphabet X.
We can now define the *Wadge class* of a set L:

Definition 21. *Let L be a subset of X^ω. The Wadge class of L is :*

$$[L] = \{L' \mid L' \subseteq Y^\omega \text{ for a finite alphabet } Y \text{ and } L' \leq_W L\}.$$

Recall that each *Borel class* $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$ is a *Wadge class*. A set $L \subseteq X^\omega$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively $\mathbf{\Pi}^0_\alpha$)-*complete set* iff for any set $L' \subseteq Y^\omega$, L' is in $\mathbf{\Sigma}^0_\alpha$ (respectively $\mathbf{\Pi}^0_\alpha$) iff $L' \leq_W L$.

There is a close relationship between Wadge reducibility and games which we now introduce.

Definition 22. *Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$. The Wadge game $W(L, L')$ is a game with perfect information between two players, player 1 who is in charge of L and player 2 who is in charge of L'.*
Player 1 first writes a letter $a_1 \in X$, then player 2 writes a letter $b_1 \in Y$, then player 1 writes a letter $a_2 \in X$, and so on.
The two players alternatively write letters a_n of X for player 1 and b_n of Y for player 2.
After ω steps, the player 1 has written an ω-word $a \in X^\omega$ and the player 2 has written an ω-word $b \in Y^\omega$. The player 2 is allowed to skip, even infinitely often, provided he really writes an ω-word in ω steps.
The player 2 wins the play iff $[a \in L \leftrightarrow b \in L']$, i.e. iff :

$[(a \in L$ and $b \in L')$ or $(a \notin L$ and $b \notin L'$ and b is infinite$)]$.

Recall that a strategy for player 1 is a function $\sigma : (Y \cup \{s\})^\star \rightarrow X$. And a strategy for player 2 is a function $f : X^+ \rightarrow Y \cup \{s\}$.

σ is a winning stategy for player 1 iff he always wins a play when he uses the strategy σ, i.e. when the nth letter he writes is given by $a_n = \sigma(b_1 \ldots b_{n-1})$, where b_i is the letter written by player 2 at step i and $b_i = s$ if player 2 skips at step i.

A winning strategy for player 2 is defined in a similar manner.

Martin's Theorem states that every Gale-Stewart Game $G(X)$ (see [Kec95]), with X a borel set, is determined and this implies the following :

Theorem 23 (Wadge). *Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$ be two Borel sets, where X and Y are finite alphabets. Then the Wadge game $W(L, L')$ is determined : one of the two players has a winning strategy. And $L \leq_W L'$ iff the player 2 has a winning strategy in the game $W(L, L')$.*

Theorem 24 (Wadge). *Up to the complement and $\equiv_W$, the class of Borel subsets of X^ω, for a finite alphabet X, is a well ordered hierarchy. There is an ordinal $|WH|$, called the length of the hierarchy, and a map d_W^0 from WH onto $|WH| - \{0\}$, such that for all $L, L' \subseteq X^\omega$:*
$d_W^0 L < d_W^0 L' \leftrightarrow L <_W L'$ and
$d_W^0 L = d_W^0 L' \leftrightarrow [L \equiv_W L'$ or $L \equiv_W L'^-]$.

The Wadge hierarchy of Borel sets of **finite rank** has length $^1\varepsilon_0$ where $^1\varepsilon_0$ is the limit of the ordinals α_n defined by $\alpha_1 = \omega_1$ and $\alpha_{n+1} = \omega_1^{\alpha_n}$ for n a non negative integer, ω_1 being the first non countable ordinal. Then $^1\varepsilon_0$ is the first fixed point of the ordinal exponentiation of base ω_1. The length of the Wadge hierarchy of Borel sets in $\mathbf{\Delta}_\omega^0 = \mathbf{\Sigma}_\omega^0 \cap \mathbf{\Pi}_\omega^0$ is the ω_1^{th} fixed point of the ordinal exponentiation of base ω_1, which is a much larger ordinal. The length of the whole Wadge hierarchy of Borel sets is a huge ordinal, with regard to the ω_1^{th} fixed point of the ordinal exponentiation of base ω_1. It is described in [Wad83,Dup01] by the use of the Veblen functions.

It is natural to ask for the Wadge hierarchy of classes of ω-languages accepted by finite machines, like **X**-automata. The Wadge hierarchy of regular ω-languages, now called the Wagner hierarchy, has been effectively determined by Wagner; it has length ω^ω [Wag79,Sel95,Sel98]. Wilke and Yoo proved in [WY95] that one can compute in polynomial time the Wadge degree of an ω-regular language. The Wadge hierarchy of ω-languages accepted by Muller *deterministic* one blind (i. e. without zero-test) counter automata is an effective extension of the Wagner hierarchy studied in [Fin01a]. Its extension to *deterministic* context free ω-languages has been determined by Duparc, its length is $\omega^{(\omega^2)}$ [DFR01,Dup03] but we do not know yet whether it is effective. Selivanov has recently determined the Wadge hierarchy of ω-languages accepted by *deterministic* Turing machines; its length is $(\omega_1^{\mathrm{CK}})^\omega$ [Sel03a,Sel03b].

In previous papers we have inductively constructed, using the work of Du-parc on the Wadge hierarchy of Borel sets [Dup01], some $\mathbf{\Delta}^0_\omega$ context free ω-languages in ε_ω Wadge degrees, where ε_ω is the ωth fixed point of the ordinal exponentiation of base ω, and also some $\mathbf{\Sigma}^0_\omega$-complete context free ω-languages, [Fin01c,Fin01b,Fin03a,Fin03b]. Notice that the Wadge hierarchy of *non-deterministic* context-free ω-languages is not effective.

We are going to show here the very surprising following result, which extends Theorem 19.

Theorem 25. *The Wadge hierarchy of the class $\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega$, hence also of the class $\mathbf{CFL}_\omega$, or of every class $\mathcal{C}$ such that $\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega \subseteq \mathcal{C} \subseteq \Sigma^1_1$, is the Wadge hierarchy of the class Σ^1_1 of ω-languages accepted by Turing machines with a Büchi acceptance condition.*

To prove this result, we are going to consider first non self dual sets. We recall the definition of Wadge degrees introduced by Duparc in [Dup01] and which is a slight modification of the previous one.

Definition 26.

(a) $d_w(\emptyset) = d_w(\emptyset^-) = 1$
(b) $d_w(L) = sup\{d_w(L') + 1 \mid L' \text{ non self dual and } L' <_W L\}$
    (for either L self dual or not, $L >_W \emptyset$).

We are going now to introduce the operation of sum of sets of infinite words which has as counterpart the ordinal addition over Wadge degrees.

Definition 27 (Wadge, see [Dup01]). *Assume that $X \subseteq Y$ are two finite alphabets, $Y - X$ containing at least two elements, and that $\{X_+, X_-\}$ is a partition of $Y - X$ in two non empty sets. Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, then*

$$L' + L =_{df} L \cup \{u.a.\beta \mid u \in X^\star, (a \in X_+ \text{ and } \beta \in L') \text{ or } (a \in X_- \text{ and } \beta \in L'^-)\}$$

This operation is closely related to the *ordinal sum* as it is stated in the following:

Theorem 28 (Wadge, see [Dup01]). *Let $X \subseteq Y$, $Y - X$ containing at least two elements, $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$ be non self dual Borel sets. Then $(L + L')$ is a non self dual Borel set and $d_w(L' + L) = d_w(L') + d_w(L)$.*

A player in charge of a set $L' + L$ in a Wadge game is like a player in charge of the set L but who can, at any step of the play, erase his previous play and choose to be this time in charge of L' or of L'^-. Notice that he can do this only one time during a play. We shall use this property below.

Lemma 29. *Let $L \subseteq \Sigma^\omega$ be a non self dual Borel set such that $d_w(L) \geq \omega$. Then it holds that $L \equiv_W \emptyset + L$.*

Notice that in the above lemma, $\emptyset$ is viewed as the empty set over an alphabet Γ such that $\Sigma \subseteq \Gamma$ and cardinal $(\Gamma - \Sigma) \geq 2$.

Proof. Assume that $L \subseteq \Sigma^\omega$ is a non self dual Borel set and that $d_w(L) \geq \omega$. We know that $\emptyset$ is a non self dual Borel set and that $d_w(\emptyset) = 1$. Thus, by Theorem 28, it holds that $d_w(\emptyset + L) = d_w(\emptyset) + d_w(L) = 1 + d_w(L)$. But by hypothesis $d_w(L) \geq \omega$ and this implies that $1 + d_w(L) = d_w(L)$. So we have proved that $d_w(\emptyset + L) = d_w(L)$.
On the other hand L is non self dual and $d_w(\emptyset + L) = d_w(L)$ imply that only two cases may happen : either $\emptyset + L \equiv_W L$ or $\emptyset + L \equiv_W L^-$.
But it is easy to see that $L \leq_W \emptyset + L$. For that purpose consider the Wadge game $W(L, \emptyset + L)$. Player 2 has clearly a winning strategy which consists in copying the play of Player 1 thus $L \leq_W \emptyset + L$. This implies that $\emptyset + L \equiv_W L^-$ cannot hold so $\emptyset + L \equiv_W L$. $\qquad\square$

Lemma 30. *Let $L \subseteq \Sigma^\omega$ be a non self dual Borel set acccepted by a Turing machine with a Büchi acceptance condition. Then there is an ω-language $L' \in$ **r-BCL**$(8)_\omega$ such that $L \equiv_W L'$.*

Proof. It is well known that there are regular ω-languages of every finite Wadge degree, [Sta97,Sel98]. These ω-languages are Boolean combinations of open sets. So we have only to consider the case of non self dual Borel sets of Wadge degrees greater than or equal to ω.

Let then $L \subseteq \Sigma^\omega$ be a non self dual Borel set acccepted by a Turing machine with a Büchi acceptance condition (in particular L is in the class **BCL**$(2)_\omega$) such that $d_w(L) \geq \omega$.
Lemma 7 states that there exists an integer $S \geq 1$ such that $\theta_S(L)$ is in the class **r-BCL**$(8)_\omega$, where E is a new letter not in Σ and $\theta_S : \Sigma^\omega \to (\Sigma \cup \{E\})^\omega$ is the function defined, for all $x \in \Sigma^\omega$, by:

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\ldots$$

We are going to prove that $\theta_S(L) \equiv_W L$.

Firstly, it is easy to see that $L \leq_W \theta_S(L)$. In order to prove this we can consider the Wadge game $W(L, \theta_S(L))$. It is easy to see that Player 2 has a winning strategy in this game which consists in copying the play of Player 1, except that Player 2 adds letters E in such a way that he has written the initial word $x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}$ while Player 1 has written the initial word $x(1).x(2).x(3).x(4)\ldots x(n)$. Notice that one can admit that a player writes a finite word at each step of the play instead of a single letter. This does not change the winner of a Wadge game.

To prove that $\theta_S(L) \leq_W L$, it suffices to prove that $\theta_S(L) \leq_W \emptyset + L$ because Lemma 29 states that $\emptyset + L \equiv_W L$. Consider the Wadge game $W(\theta_S(L), \emptyset + L)$. Player 2 has a winning strategy in this play which consists first in copying the

play of player 1 except that Player 2 skips when player 1 writes a letter E. He continues forever with this strategy if the word written by player 1 is always a prefix of some ω-word of $\theta_S(\Sigma^\omega)$. Then after ω steps Player 1 has written an ω-word $\theta_S(x)$ for some $x \in \Sigma^\omega$, and Player 2 has written x. So in that case $\theta_S(x) \in \theta_S(L)$ iff $x \in L$ iff $x \in \emptyset + L$.

But if at some step of the play, Player 1 "goes out of" the closed set $\theta_S(\Sigma^\omega)$ because the word he has now written is not a prefix of any ω-word of $\theta_S(\Sigma^\omega)$, then its final word will be surely outside $\theta_S(\Sigma^\omega)$ hence also outside $\theta_S(L)$. Player 2 can now writes a letter of $\Gamma - \Sigma$ in such a way that he is now like a player in charge of the emptyset and he can now writes an ω-word u so that his final ω-word will be outside $\emptyset + L$. Thus Player 2 wins this play too.

Finally we have proved that $L \leq_W \theta_S(L) \leq_W L$ thus it holds that $\theta_S(L) \equiv_W L$. This ends the proof.

$\square$

Lemma 31. *Let $L \subseteq \Sigma^\omega$ be a non self dual Borel set in the class $\mathbf{r}\text{-}\mathbf{BCL}(8)_\omega$. Then there is an ω-language $L' \in \mathbf{r}\text{-}\mathbf{BCL}(1)_\omega$ such that $L \equiv_W L'$.*

Proof. As in the preceding proof we can consider only ω-languages of Wadge degrees greater than or equal to ω.

Let then $L = L(\mathcal{A}) \subseteq \Sigma^\omega$ be a non self dual Borel set accepted by a real time 8-counter Büchi automaton $\mathcal{A}$ such that $d_w(L) \geq \omega$. We have shown in the preceding section that $\phi_K(h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^-)$ is in the class $\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega$, where h_K is the continuous mapping $h_K : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$ defined by : for all $x \in \Sigma^\omega$,

$$h_K(x) = A.0^K.x(1).B.0^{K^2}.A.0^{K^2}.x(2).B.0^{K^3}.A.0^{K^3}.x(3).B \ldots B.0^{K^n}.A.0^{K^n}.x(n).B \ldots$$

and the mapping $\phi_K : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ is defined by: for all $x \in (\Sigma \cup \{A, B, 0\})^\omega$,

$$\phi_K(x) = F^{K-1}.x(1).F^{K-1}.x(2).F^{K-1}.x(3) \ldots F^{K-1}.x(n).F^{K-1}.x(n+1).F^{K-1} \ldots$$

We can now prove, by a very similar reasoning as in the proof of the preceding Lemma 30, using the fact that $d_w(L) \geq \omega$, that

$$L \equiv_W h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^- \equiv_W \phi_K(h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^-)$$

But $\phi_K(h_K(L(\mathcal{A})) \cup h_K(\Sigma^\omega)^-)$ is in the class $\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega$, and this ends the proof.

$\square$

End of Proof of Theorem 25. Let $L \subseteq \Sigma^\omega$ be a Borel set accepted by a Turing machine with a Büchi acceptance condition (in particular L is in the class $\mathbf{BCL}(2)_\omega$). If the Wadge degree of L is finite, it is well known that it is Wadge equivalent to a regular ω-language, hence also to an ω-language in the

class $\mathbf{r\text{-}BCL}(1)_\omega$. If L is non self dual and its Wadge degree is greater than or equal to ω, then we can infer from Lemmas 30 and 31 that there is an ω-language $L' \in \mathbf{r\text{-}BCL}(1)_\omega$ such that $L \equiv_W L'$.

It remains to consider the case of self dual Borel sets. The alphabet Σ being finite, a self dual Borel set L is always Wadge equivalent to a Borel set in the form $\Sigma_1.L_1 \cup \Sigma_2.L_2$, where (Σ_1, Σ_2) form a partition of Σ, and $L_1, L_2 \subseteq \Sigma^\omega$ are non self dual Borel sets such that $L_1 \equiv_W L_2^-$. Moreover L_1 and L_2 can be taken in the form $L_{(u_1)} = u_1.\Sigma^\omega \cap L$ and $L_{(u_2)} = u_2.\Sigma^\omega \cap L$ for some $u_1, u_2 \in \Sigma^\star$, see [Dup03]. So if $L \subseteq \Sigma^\omega$ is a self dual Borel set accepted by a Turing machine with a Büchi acceptance condition then $L \equiv_W \Sigma_1.L_1 \cup \Sigma_2.L_2$, where (Σ_1, Σ_2) form a partition of Σ, and $L_1, L_2 \subseteq \Sigma^\omega$ are non self dual Borel sets accepted by a Turing machine with a Büchi acceptance condition. We have already proved that there is an ω-language $L'_1 \in \mathbf{r\text{-}BCL}(1)_\omega$ such that $L'_1 \equiv_W L_1$ and an ω-language $L'_2 \in \mathbf{r\text{-}BCL}(1)_\omega$ such that $L_2^{'-} \equiv_W L_2$. Thus $L \equiv_W \Sigma_1.L_1 \cup \Sigma_2.L_2 \equiv_W \Sigma_1.L'_1 \cup \Sigma_2.L'_2$ and $\Sigma_1.L'_1 \cup \Sigma_2.L'_2$ is in the class $\mathbf{r\text{-}BCL}(1)_\omega$. $\square$

Remark 32. *We have only considered above the Wadge hierarchy of* **Borel sets**. *If we assume the axiom of $\mathbf{\Sigma}_1^1$-determinacy, then Theorem 24 can be extended by considering the class of analytic sets instead of the class of Borel sets. In fact in that case any set which is analytic but not Borel is $\mathbf{\Sigma}_1^1$-complete, see [Kec95]. So there is only one more Wadge degree containing $\mathbf{\Sigma}_1^1$-complete sets. It was already proved in [Fin03a] that there is a $\mathbf{\Sigma}_1^1$-complete set accepted by a Büchi 1-counter automaton and it is easy to see from the proof that one can find such a $\mathbf{\Sigma}_1^1$-complete set accepted by a Büchi 1-counter real-time automaton.*

Remark 33. *The result given by Theorem 19 can now be deduced from Theorem 25 and it can be seen as a particular case of this last result, because the Wadge hierarchy is a refinement of the Borel hierarchy and, for each countable non null ordinal γ, $\mathbf{\Sigma}_\gamma^0$-complete sets (respectively, $\mathbf{\Pi}_\gamma^0$-complete sets) form a single equivalence class of $\equiv_W$, i.e. a single Wadge degree, [Kec95]. However we have preferred to expose the results given in this paper by considering firstly the Borel hierarchy. This way the reader who is just interested by the Borel hierarchy of ω-languages can read this part and skip Section 6 about the Wadge hierarchy.*

7 Concluding remarks

We have proved that the Borel and the Wadge hierarchies of classes $\mathbf{r\text{-}BCL}(1)_\omega$ and $\mathbf{CFL}_\omega$ are also the Borel and the Wadge hierarchies of the class Σ_1^1. The methods used in this paper are different from those used in previous papers on context free ω-languages [Fin01c,Fin01b,Fin03a,Fin03b], where we gave an inductive construction of some $\mathbf{\Delta}_\omega^0$ context free ω-languages of a given Borel rank or Wadge degree, using work of Duparc on the Wadge hierarchy of $\mathbf{\Delta}_\omega^0$ Borel sets, [Dup01]. However it will be possible to combine both methods for the effective construction of ω-languages in the class $\mathbf{r\text{-}BCL}(1)_\omega$, and of 1-counter Büchi automata accepting them, of a given Wadge degree among the ε_ω degrees obtained in [Fin01b] for $\mathbf{\Delta}_\omega^0$ context free ω-languages.

Finally we mention that in another paper, using the results of this paper and applying similar methods to the study of topological properties of infinitary rational relations, we prove that their Wadge and Borel hierarchies are equal to the corresponding hierarchies of the classes $\mathbf{r\text{-}BCL}(1)_\omega$, $\mathbf{CFL}_\omega$ or Σ_1^1, [Fin06].

Acknowledgements

Thanks to the anonymous referee for useful comments on a preliminary version of this paper.

References

ABB96. J.-M. Autebert, J. Berstel, and L. Boasson. Context free languages and push-down automata. In *Handbook of formal languages, Vol. 1*. Springer-Verlag, 1996.

Ber79. J. Berstel. *Transductions and context free languages*. Teubner Studienbücher Informatik, 1979.

Cac02. T. Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *Proceedings of the International Conference ICALP 2002*, volume 2380 of *Lecture Notes in Computer Science*, pages 704–715. Springer, 2002.

CDT03. T. Cachat, J. Duparc, and W. Thomas. Solving pushdown games with a Σ_3-winning condition. In *Proceedings of the 11th Annual Conference of the European Association for Computer Science Logic, CSL 2002*, volume 2471 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2003.

CG77. R.S. Cohen and A.Y. Gold. Theory of ω-languages, parts one and two. *Journal of Computer and System Science*, 15:169–208, 1977.

CG78. R.S. Cohen and A.Y. Gold. ω-computations on deterministic pushdown machines. *Journal of Computer and System Science*, 16:275–300, 1978.

CP97. O. Carton and D. Perrin. Chains and superchains for ω-rational sets, automata and semigroups. *International Journal of Algebra and Computation*, 7(7):673–695, 1997.

CP99. O. Carton and D. Perrin. The Wagner hierarchy of ω-rational sets. *International Journal of Algebra and Computation*, 9(5):597–620, 1999.

DFR01. J. Duparc, O. Finkel, and J.-P. Ressayre. Computer science and the fine structure of Borel sets. *Theoretical Computer Science*, 257(1–2):85–105, 2001.

DR05. J. Duparc and M. Riss. The missing link for ω-rational sets, automata, and semigroups. *to appear in International Journal of Algebra and Computation*, 2005.

Dup01. J. Duparc. Wadge hierarchy and Veblen hierarchy: Part 1: Borel sets of finite rank. *Journal of Symbolic Logic*, 66(1):56–86, 2001.

Dup03. J. Duparc. A hierarchy of deterministic context free ω-languages. *Theoretical Computer Science*, 290(3):1253–1300, 2003.

EH93. J Engelfriet and H. J. Hoogeboom. X-automata on ω-words. *Theoretical Computer Science*, 110(1):1–51, 1993.

Fin01a. O. Finkel. An effective extension of the Wagner hierarchy to blind counter automata. In *Proceedings of Computer Science Logic, 15th International Workshop, CSL 2001*, volume 2142 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2001.

Fin01b.   O. Finkel. On the Wadge hierarchy of omega context free languages. In *Proceedings of the International Workshop on Logic and Complexity in Computer Science, held in Honor of Anatol Slissenko for his 60th Birthday*, pages 69–79, Créteil, France, 2001.

Fin01c.   O. Finkel. Topological properties of omega context free languages. *Theoretical Computer Science*, 262(1–2):669–697, 2001.

Fin01d.   O. Finkel. Wadge hierarchy of omega context free languages. *Theoretical Computer Science*, 269(1–2):283–315, 2001.

Fin03a.   O. Finkel. Borel hierarchy and omega context free languages. *Theoretical Computer Science*, 290(3):1385–1405, 2003.

Fin03b.   O. Finkel. On omega context free languages which are Borel sets of infinite rank. *Theoretical Computer Science*, 299(1-3):327–346, 2003.

Fin05.   O. Finkel. Borel ranks and Wadge degrees of context free ω-languages. In *Proceedings of New Computational Paradigms: First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands*, volume 3526 of *Lecture Notes in Computer Science*, pages 129–138. Springer, 2005.

Fin06.   O. Finkel. On the accepting power of two-tape Büchi automata. In *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science, STACS 2006*, volume 3884 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 2006.

FS03.   O. Finkel and P. Simonnet. Topology and ambiguity in omega context free languages. *Bulletin of the Belgian Mathematical Society*, 10(5):707–722, 2003.

HU79.   J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.

Jec02.   T. Jech. *Set theory, third edition*. Springer, 2002.

Kec95.   A. S. Kechris. *Classical descriptive set theory*. Springer-Verlag, New York, 1995.

KMS89.   A. S. Kechris, D. Marker, and R. L. Sami. Π_1^1 Borel sets. *Journal of Symbolic Logic*, 54(3):915–920, 1989.

Lin75.   M. Linna. On ω-words and ω-computations. *Ann. Univ. Turku. Ser A I*, 168:53, 1975.

Lou05.   A. Louveau. Personal communication. 2005.

LT94.   H. Lescow and W. Thomas. Logical specifications of infinite computations. In J. W. de Bakker, Willem P. de Roever, and Grzegorz Rozenberg, editors, *A Decade of Concurrency*, volume 803 of *Lecture Notes in Computer Science*, pages 583–621. Springer, 1994.

Mos80.   Y. N. Moschovakis. *Descriptive set theory*. North-Holland Publishing Co., Amsterdam, 1980.

Niv77.   M. Nivat. Mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique et Applications*, 11:311–327, 1977.

Niv78.   M. Nivat. Sur les ensembles de mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique et Applications*, 12(3):259–278, 1978.

PP04.   D. Perrin and J.-E. Pin. *Infinite words, automata, semigroups, logic and games*, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004.

Sel95.   V.L. Selivanov. Fine hierarchy of regular ω-languages. In *Proceedings of the International Joint Conference on the Theory and Practice of Software Development TAPSOFT-95, in Aarhus, Denmark*, volume 915 of *Lecture Notes in Computer Science*, pages 277–287. Springer, 1995.

Sel98.   V.L. Selivanov. Fine hierarchy of regular ω-languages,. *Theoretical Computer Science*, 191:37–59, 1998.

Sel03a.   V.L. Selivanov. Wadge degrees of ω-languages of deterministic Turing machines. In *Proceedings of the International Conference STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany*, volume 2607 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2003.

Sel03b.   V.L. Selivanov. Wadge degrees of ω-languages of deterministic Turing machines. *RAIRO-Theoretical Informatics and Applications*, 37(1):67–83, 2003.

Ser04a.   O. Serre. *Contribution à l'étude des jeux sur des graphes de processus à pile*. PhD thesis, Université Paris VII, 2004.

Ser04b.   O. Serre. Games with winning conditions of high Borel complexity. In *Proceedings of the International Conference ICALP 2004*, volume 3142 of *Lecture Notes in Computer Science*, pages 1150–1162. Springer, 2004.

Sim92.   P. Simonnet. *Automates et théorie descriptive*. PhD thesis, Université Paris VII, 1992.

Sta86.   L. Staiger. Hierarchies of recursive ω-languages. *Elektronische Informationsverarbeitung und Kybernetik*, 22(5-6):219–241, 1986.

Sta87.   L. Staiger. Research in the theory of ω-languages. *Journal of Information Processing and Cybernetics*, 23(8-9):415–439, 1987. Mathematical aspects of informatics (Mägdesprung, 1986).

Sta97.   L. Staiger. ω-languages. In *Handbook of formal languages, Vol. 3*, pages 339–387. Springer, Berlin, 1997.

SW74.   L. Staiger and K. Wagner. Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. *Elektron. Informationsverarbeit. Kybernetik*, 10:379–392, 1974.

Tho90.   W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, Formal models and semantics, pages 135–191. Elsevier, 1990.

Tho02.   W. Thomas. Infinite games and verification (extended abstract of a tutorial). In *Proceedings of the International Conference on Computer Aided Verification CAV 2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 58–64. Springer, 2002.

Wad83.   W. Wadge. *Reducibility and determinateness in the Baire space*. PhD thesis, University of California, Berkeley, 1983.

Wag79.   K. Wagner. On ω-regular sets. *Information and Control*, 43(2):123–177, 1979.

Wal00.   I. Walukiewicz. Pushdown processes: games and model checking. *Information and Computation*, 157:234–263, 2000.

WY95.   T. Wilke and H. Yoo. Computing the Wadge degree, the Lifschitz degree, and the Rabin index of a regular language of infinite words in polynomial time. In P.D. Mosses, M. Nielsen, and M.I Schwartzbach, editors, *TAPSOFT 95*, volume 915 of *Lect. Notes in Comp. Sci.*, pages 288–302. Springer Verlag, Berlin, Heidelberg, New York, 1995.